**1、思考：什么时候使用this关键字？**

**什么时候使用private？**

**如何给成员变量赋值？**

答案：

使用this关键字：

创建对象，为对象中的成员赋值时使用

成员变量与局部变量重名时使用

使用private关键字：

封装类中成员时使用，隐藏实现细节

给成员变量赋值：

通过构造方法赋值

通过setXxx()方法赋值

1. **成员变量与局部变量的区别？**

答案：

A:在类中的位置不同

成员变量：在类中方法外

局部变量：在方法定义中或者方法声明上

B:在内存中的位置不同

成员变量：在堆内存

局部变量：在栈内存

C:生命周期不同

成员变量：随着对象的创建而存在，随着对象的消失而消失

局部变量：随着方法的调用而存在，随着方法的调用完毕而消失

D:初始化值不同

成员变量：有默认初始化值

局部变量：没有默认初始化值，必须定义，赋值，然后才能使用。

1. **什么是匿名对象？什么时候使用匿名对象？**

答案：

匿名对象指：没有起名字的对象

使用匿名对象:

a:调用方法，仅仅只调用一次的时候

b:匿名对象可以作为实际参数传递

1. **private关键字在哪里使用？ 被private修饰的成员有什么特点？**

答案：

类中的成员需要私有的时候使用private关键字

特点：

是一个权限修饰符。

可以修饰成员(成员变量和成员方法)

被private修饰的成员只在本类中才能访问

1. **方法重写和重载有什么区别?**

答案：

重载Overload表示同一个类中可以有多个名称相同的方法，但这些方法的参数列表各不相同(即参数个数或类型不同)。

重写Override表示子类中的方法可以与父类的某个方法的名称和参数完全相同，通过子类创建的实例对象调用这个方法时，将调用子类中的定义方法，这相当于把父类中定义的那个完全相同的方法给覆盖了，这也是面向对象编程的多态性的一种表现。子类覆盖父类的方法时，只能比父类抛出更少的异常，或者是抛出父类抛出的异常的子异常，因为子类可以解决父类的一些问题，不能比父类有更多的问题。子类方法的访问权限只能比父类的更大，不能更少。如果父类的方法是private类型，那么，子类则不存在覆盖的限制，相当于子类中增加了一个全新的方法。

**写出这个输出的结果**

**class Test {**

**int num = 3;**

**public static void main(String[] args) {**

**Test t = new Test();//0x11 3 5 6**

**t.num = 5;**

**method(t);//0x11**

**System.out.println(t.num);**

**}**

**public static void method(Test t){//0x11 6**

**t.num = 6;**

**}**

**}**

答案：6

**2、写出这个输出的结果**

**class Test {**

**int num = 3;**

**public static void main(String[] args) {**

**Test t = new Test();// 0x11 3 5**

**t.num = 5;**

**method(new Test());//0x22**

**System.out.println(t.num);**

**}**

**public static void method(Test t){//0x22 6**

**t.num = 6;**

**}**

**}**

答案：5

1. **请补全下面程序中横线位置的代码(可以编写多行代码)**

**public class \_\_\_\_\_\_\_\_\_\_\_\_{**

**int x, y;**

**public Myclass( int i, \_\_\_\_\_\_\_) {**

**x=i;**

**y=j;**

**}**

**}**

答案：

位置一：Myclass

位置二：int j

1. **请写出程序的运行结果**

**class Fu {**

**public int num = 10;**

**public Fu() {**

**System.out.println("fu");**

**}**

**}**

**class Zi {**

**public int num = 20;**

**public Zi() {**

**System.out.println("zi");**

**}**

**public void show() {**

**int num = 30;**

**System.out.println(num);**

**System.out.println(this.num);**

**System.out.println(super.num);**

**}**

**}**

**class Test {**

**public static void main(String[] args) {**

**Zi z = new Zi();**

**z.show();**

**}**

**}**

答案：编译出错， super.num 无法访问

需要继承父类后才能访问num

class Zi extends Fu

1. **请写出程序的运行结果**

**class Demo {**

**public void method(int age){**

**System.out.println("int age");**

**}**

**public void method(String name){**

**System.out.println("String name");**

**}**

**public void method(String name, int age){**

**System.out.println(name +"---"+ age);**

**}**

**}**

**public class Test{**

**public static void main(String[] args){**

**Demo d = new Demo();**

**d.method("小明");**

**d.method(18);**

**}**

**}**

答案：String name

int age

1. **给出以下代码，请问该程序的运行结果是什么？**

**class Demo {**

**String s = "Outer";**

**public static void main (String[] args) {**

**S2 s2 = new S2();**

**s2.display();**

**}**

**}**

**class S1 {**

**String s = "S1";**

**public void display(){**

**System.out.println(s);**

**}**

**}**

**class S2 extends S1 {**

**String s = "S2";**

**}**

答案：S1

1. **给出以下代码，请问该程序的运行结果是什么？**

**class S1 {**

**public static void main(String[] args) {**

**new S2();**

**}**

**public S1(){**

**System.out.println("S1");**

**}**

**}**

**class S2 extends S1 {**

**public S2 () {**

**System.out.println("S2");**

**}**

**}**

答案：S1

**S2**